

11/01/00
JC946 U.S. PTO

11-02-00

A

FISH & RICHARDSON P.C.

225 Franklin Street
Boston, Massachusetts
02110-2804

Telephone
617 542-5070

Facsimile
617 542-8906

Web Site
www.fr.com

Frederick P. Fish
1855-1930

W.K. Richardson
1859-1951

November 1, 2000

Attorney Docket No.: 09765-020001

Box Patent Application
Commissioner for Patents
Washington, DC 20231

Presented for filing is a new patent application claiming priority from a provisional patent application of:

Applicant: JAMES N. REES

Title: TRANSMITTING INFORMATION OVER A NETWORK

Enclosed are the following papers, including those required to receive a filing date under 37 CFR §1.53(b):

	<u>Pages</u>
Specification	9
Claims	3
Abstract	1
Declaration	[To be Filed at a Later Date]
Drawing(s)	10

Enclosures:
— Postcard.

Under 35 USC §119(e)(1), this application claims the benefit of prior U.S. provisional application 60/162,875, filed November 1, 1999.

CERTIFICATE OF MAILING BY EXPRESS MAIL

Express Mail Label No EL224703503US

I hereby certify under 37 CFR §1.10 that this correspondence is being deposited with the United States Postal Service as Express Mail Post Office to Addressee with sufficient postage on the date indicated below and is addressed to the Commissioner for Patents, Washington, D.C. 20231.

11-01-2000
Date of Deposit

Victor J. Mahoney
Signature

Victor J. Mahoney
Typed or Printed Name of Person Signing Certificate

BOSTON
DALLAS
DELAWARE
NEW YORK
SAN DIEGO
SILICON VALLEY
TWIN CITIES
WASHINGTON, DC

225 Franklin Street
Boston, Massachusetts
02110-2804
Telephone
617 542-5070
Facsimile
617 542-8906
Web Site
www.fr.com
JC932 U.S. PTO
09/704221
11/01/00

Commissioner for Patents
November 1, 2000
Page 2

Basic filing fee	\$710
Total claims in excess of 20 times \$18	\$18
Independent claims in excess of 3 times \$80	\$0
Fee for multiple dependent claims	\$0
Total filing fee:	\$728

Under 37 CFR §1.53(f), no filing fee is being paid at this time.


If this application is found to be incomplete, or if a telephone conference would otherwise be helpful, please call the undersigned at (617) 542-5070.

Kindly acknowledge receipt of this application by returning the enclosed postcard.

Please send all correspondence to:

DENIS G. MALONEY
Fish & Richardson P.C.
225 Franklin Street
Boston, MA 02110-2804

Respectfully submitted,



Denis G. Maloney
Reg. No. 29,670
Enclosures
DGM/laf/mgc
20172041.doc

001111 : 11111111

APPLICATION
FOR
UNITED STATES LETTERS PATENT

TITLE: TRANSMITTING INFORMATION OVER A NETWORK
APPLICANT: JAMES N. REES

CERTIFICATE OF MAILING BY EXPRESS MAIL

Express Mail Label No. EL224703503US

I hereby certify under 37 CFR §1.10 that this correspondence is being deposited with the United States Postal Service as Express Mail Post Office to Addressee with sufficient postage on the date indicated below and is addressed to the Commissioner for Patents, Washington, D.C. 20231.

11-01-2000
Date of Deposit

Victor J. Mahoney
Signature

Victor J. Mahoney
Typed or Printed Name of Person Signing Certificate

TRANSMITTING INFORMATION OVER A NETWORK

TECHNICAL FIELD

This invention relates to transmitting information over a network.

BACKGROUND

Network servers, such as web servers, normally receive requests for information from remote client computers over a network, such as the Internet or an intranet. For example, the requests for information may be HyperText Transfer Protocol (HTTP) "POST" or "GET" requests that are transmitted by a web browser on a client computer when a user submits a query to a web server. Network servers may transmit information to the client computer in response to the request. The transmitted information may be data such as a web page that is formatted in the HyperText Markup Language (HTML), promulgated by the World Wide Web Consortium (W3C). The transmitted information may also be an executable that is, for example, written in the JAVA™ programming language, by Sun Microsystems Inc. A JAVA™ interpreter on the client computer may execute such an executable.

It is sometimes necessary to transmit an executable to the client computer along with data that is to be processed on the client computer by the executable. For example, a web server may respond to a request for flight itineraries by sending the flight itinerary data along with a JAVA™ applet for graphically displaying the data.

SUMMARY

A first general aspect of the invention relates to a method that includes transmitting information by transmitting an executable to a remote location over a network, generating data, and transmitting the generated data to the remote location over the network. At least part of the data is generated while the executable is being transmitted.

By generating part of the data while the executable is being transmitted, the method reduces the delay or latency before the executable and the data arrive at the remote location. The reduction in the delay makes the method more responsive to a user.

A second general aspect of the invention relates to an article that includes a machine-readable medium, which stores machine-executable instructions operable to cause a machine to perform the method of the first general aspect of the invention.

A third general aspect of the invention relates to an apparatus that includes a storage system that stores the executable and the instructions, and a server processor, which executes the instructions to perform the method of the first general aspect of the invention.

Embodiments of the invention may include one or more of the following features. A client at the remote location receives the transmitted executable and then later receives the generated data. The client includes a client processor that executes the executable and indicates that the client computer is waiting for the data. By receiving and executing the executable before the data is received, the client can inform the user that the client is waiting for the data, further increasing the responsiveness of the method to the user. When the data is received, the client processor executes the executable to process the data.

A first portion of the data is generated before a second portion of the data and at least part of the first portion of the data is transmitted while the second portion is being generated, further reducing the latency of the method. The client receives the first portion of the data before it receives the second portion of the data and the executable processes the first portion of the data before the second portion is received. The executable also causes the client to indicate that the first portion of the data has been received before the second portion is received, thereby further increasing the responsiveness of the client.

The information is transmitted in response to a request for information, while the data is transmitted to the client in response to a request for data separate from the request for information. A web page containing a directive to a web browser to request the executable is transmitted to the remote location and a request for the executable is later received from the web browser. The executable is transmitted to the remote location in response to the request for the executable.

Among other advantages of the invention, the reduction in latency and the improved responsiveness of the client improve the experience of the user.

The details of one or more embodiments of the invention are set forth in the accompanying drawings and the description below. Other features, objects, and advantages of the invention will be apparent from the description and drawings, and from the claims.

DESCRIPTION OF DRAWINGS

FIG. 1 is a block diagram of a network that implements the invention and includes a server and clients;

FIGS. 2-5 show web pages that are transmitted by the server of FIG. 1 to the clients of FIG. 1;

FIGS. 6A and 6B are flowcharts of processes implemented by the server of FIG. 1 and a client of FIG. 1;

FIGS. 7A-7C show segments of the HTML representation of the web page of FIG. 5; and

FIGS. 8A and 8B show flowcharts of processes implemented by alternative embodiments of the server and the client of FIG. 1.

Like reference symbols in the various drawings indicate like elements.

DETAILED DESCRIPTION

As shown in FIG. 1, a server 10 receives requests 12a, 12b for travel information 18 from remote client computers 14a, 14b over a network 16, such as the Internet or an intranet. As will be described in greater detail below, the server 10 may respond to a request 12 by transmitting travel data and a JAVA™ applet to the client over the network 16. The applet is executed on the client computer to process the travel data. The server 10 is configured to generate the travel data while transmitting the applet and then later transmit the travel data to the client computer, as will be described in greater detail below.

Server 10 includes a processor 22 and a network interface 20 that transmits information 18 and receives requests 12 from the network 16. The server 10 also includes a computer-readable storage subsystem 24 that stores computer programs, which are executed by processor 22. Storage subsystem 24 may include a memory, a CDROM, a floppy disk, a hard disk drive, a disk array, or a floppy disk. A multi-tasking operating system program 26 allows the server 10 to perform multiple tasks at the same time. A web server program 28 receives the requests 12 from a client computers 14a, 14b through the network interface 20 and responds to the requests by transmitting information 18 through the network interface 20 to the client computer. For example, web server 28 may send web pages 30 or a client executable 32, such as a JAVA™ applet, both of which may be stored in storage 28. Web

server 28 may also respond to requests by using a travel program module 34 to generate travel data 36 from a travel database management system program (DBMS), and transmit the travel data through the network interface 20.

The travel module 34 is a computer program module, such as a dynamically linked library, that is invoked by the web server 28. Web server 28 uses the multi-tasking operating system 26 to invoke the travel module 34 while the web server 28 is still running. For example, web server 28 may invoke the travel module 34 as a separate operating system process or a separate thread.

The client computer 14a includes a network interface 39 and a storage subsystem 40 that stores client programs, such as a web browser 42 and a JAVATM interpreter 44 that is associated with the web browser. Client computer 14a also includes a processor 46 that executes web browser 42 to receive input from a user through an input device 48 and transmit the input as requests 12a to the server 10 through the network interface 39. The input device 48 may be computer keyboard or a mouse associated with the client computer 14a. Web browser 42 also receives information 18 from server 10 through the network interface 39 and presents it to the user on a display 50 that is associated with the client computer 14a. When the information received includes client executable 32, web browser 42 invokes java interpreter 44 to execute the executable 32. Web browser 42 also stores the information 18 in a cache 52. Should the user later request the stored information 18, the web browser retrieves the information 18 from the cache 52 thereby eliminating the need to receive the information from the network 16. This makes the information available more quickly because the network 16 takes longer to transmit the information 18 than it takes to retrieve the information from the cache 52.

Referring to FIGS. 1 and 2, a user requests a search web page 30a for searching for travel itineraries, by typing a Uniform Resource Location (URL) 50 associated with the web page 30a into an URL input 52 associated with the web browser 42. The web browser transmits the request to the server 10 through the network 16 and the server 10 responds to the request by sending the web page 30a to the browser 42, which in turn presents the web page to the user on display 50.

A user enters search criteria for searching for travel itineraries including a point of departure, a departure time, a point of arrival and an arrival time into inputs 54a-54d

contained within search web page 30a. The user then clicks on submission button 56, causing the browser 42 to generate an HTTP post request 12 and transmit the request to the server 10 over the network 16. Upon receiving the request, the server 10 transmits the client executable 32 to the browser 42 while generating the information 18 about the travel itineraries.

As shown in FIG. 3, the browser 42 presents a web page 30b while the browser waits for information 18 about travel itineraries from the server 10. Web page 30b informs 60 the user to wait while the browser downloads client executable 32, which is a travel browser for displaying the requested information 18 about travel itineraries. Web page 30b also informs the user of the time 62 it will take to download the client executable. Upon receiving the client executable 32, web browser 42 invokes JAVATM interpreter 44 to execute the executable 32.

As shown in FIG. 4, client executable 32 presents a panel 70 on a web page 30c that informs 72 the user that the executable 32 is waiting for data 36 on travel itineraries associated with the user's request 12. When web browser 42 receives the data 36, client executable 32 processes the data, for example, by decoding, uncompressing, and preparing the data 36 for graphical presentation on the display 50.

As shown in FIG. 5, client executable 32 replaces the panel 70 (FIG. 4) on web page 30c with a new panel 74, which contains a graphical presentation 76 of such itinerary data as departure flights 76a and arrival flights 76b.

Referring to FIG. 6A, upon receiving (600) the request from browser 42, web server 28 transmits (602) a first segment of the web page 30c (FIGS. 4 and 5) to the browser 42. The first segment includes a directive to the browser 42 to request the client executable 32 from the server 32. The web server 28 invokes the travel module 34, causing the travel module to generate (604) another segment of the web page 30c containing a portion of the travel itinerary data 36. While the travel module is generating the segment of web page 30c, the web server 28 receives (612) a request for client executable 32 from browser 42, and transmits (614) client executable 32 to the browser 42. The web server 28 transmits at least part of client executable 32 while the travel module is generating the segment of the web page.

Web server 28 transmits (606) the generated segment of the web page to the browser 42 and checks (608) with the travel module 34 to determine whether there is another portion of the travel itinerary data 36 to be generated. If there is more data 36 to be generated, the web server 10 invokes the travel module 34 to generate (604) yet another segment of the web page. A part of the segment of the web page may be generated (604) while the previous segment of the web page is being transmitted.

Otherwise, if there is no more data to be generated, the server 10 terminates the process.

Referring to FIG. 6B, when the client computer 14a transmits (650) a data request query to the server 10, the client computer 14a uses web page 30b (FIG. 3) to inform (652) the user that the browser is waiting for the client executable 32 to be downloaded. When the browser 42 receives (654) the first segment of the web page 30c, the directive contained within the web page causes the web browser to transmit (656) a request for the client executable 32 from the server 10. Upon receiving (658) the client executable 32 from the server, java interpreter 44 executes the client executable 32 and informs (660) the user in panel 70 (FIG. 4) that the browser 42 is waiting for data from the server 10.

On receiving (662) another segment of the web page containing a portion of the travel itinerary data 36, client executable 32 decodes (664) the portion of the data and graphically displays (666) the decoded data in panel 74 (FIG. 5). Certain client executables 32 may also display an indication to the user that the new segment has been received. For example, they may update a display of the percentage of the travel itinerary data 36 that has been received. The client executable 32 then checks (668) whether the portion of the data 36 in the segment is the last portion, as will be described later with reference to FIG. 7C. If it is not the last portion of the data 36, the web page repeats the process (662-668) of receiving another segment of web page 30c. Otherwise, if it is the last segment, the web browser 42 terminates the process.

As shown in FIG. 7A, the first segment 80 of the web page 30c includes an HTML directive 82 to the browser 42 to request the client executable 32 from a URL 84 and invoke java interpreter 44 to execute the executable 32. Browser 42 responds to the first segment 80 by requesting the executable 32 and storing it in cache 52. The directive 82 also contains attributes that the java interpreter 44 should invoke the executable with. For instance, the

code attribute 88 indicates to the interpreter 44 that the executable 32 should be invoked with no data, attributes 90 indicate the size of the panel 70 (FIG. 4) that the executable 32 should display, while attributes 92 indicate the spacing of data displayed in the panel 70. The first segment 80 also contains a set of parameters 94, including a list 96 of the locations where the executable 32 can find such information as logos and sounds.

As shown in FIG. 7B, the second segment 100 of the web page 30c also includes an HTML directive 102 to the browser to request the client executable 32 from the URL 84. However, the browser does not retrieve the executable 32 from the server 10 because the browser 42 finds the executable in cache 32. Although the directive 102 also contains attributes 104, 106 indicating the size and spacing that the applet should be invoked with, the values of the attributes are set to zero. Consequently, the second invocation of the executable 32 does not create a new panel. Directive 102 is associated with a parameter 108 that is also contained in the second segment 100 of the web page 30c. The parameter includes a first portion 110 of the travel data 36 for display by the executable 32. As previously described, server 10 may send any number of such segments 100 of the web page each containing a portion of the data. The server 10 may be configured to transmit segments of web pages containing differently sized portions 110 of the data 36, depending on the speed and reliability of the network 16. For most networks, the server 10 is configured to transmit portions of data that are larger than the portion 110.

As shown in FIG. 7C, the last segment 120 of the web page 30c is similar to the second segment 100 (FIG. 7B), but includes an additional parameter 122, which indicates to the executable 32 that there are no more data portions to be received. The client applet 32 checks (668) (FIG. 6B) each segment of the web page 30c for the parameter 122 to determine whether the portion of the data 36 contained in the segment is the last portion of the data. Additionally, the last segment 120 contains html keywords 124 that inform the browser 42 that the browser 42 has received the entire web page 30c.

The client executable 32 graphically displays the data 36 from the segments in the panel 74 (FIG. 5).

Referring to FIG. 8A, in an alternate embodiment of the invention, upon receiving (800) the request from browser 42, web server 28 generates (802) a session identity (ID) transmits (804) a web page to the browser 42 that includes the session ID along with a

directive to the web browser 42 to request the client executable 32 from the server 10. The web server 28 invokes the travel module 34, causing the travel module to generate (806) the travel itinerary data 36. While the travel module 34 is generating the data 36, the web server 28 receives (814) a request for client executable 32 from browser 42, and transmits (816)
5 client executable 32 to the browser 42. The web server 28 transmits at least part of client executable 32 while the travel module is generating the segment of the web page.

Web server 28 associates (808) the data 36 generated by travel module 34 with the session ID and checks (810) whether a request for data that includes the session ID has been received. If the request has not been received, the web server waits (810) for the request.
10 Otherwise, if the request has been received, the web server transmits (812) the data to the requesting client computer 14a and terminates the process. The server 10 may also be configured to segment the generated data and transmit the segments in sequence.

Referring to FIG. 8B, in the alternate embodiment, when the client computer 14a transmits (850) a query to the server 10, the client computer 14a uses web page 30b (FIG. 3) to inform (852) the user that the browser is waiting for the client executable 32 to be
15 downloaded. When the browser 42 receives (854) the web page with the session ID, the directive contained within the web page causes the web browser to transmit (856) a request for the client executable 32 from the server 10. Upon receiving (858) the client executable 32 from the server, java interpreter 44 executes the client executable 32 and informs (860)
20 the user in panel 70 (FIG. 4) that the browser 42 is waiting for data from the server 10.

Client executable 32 transmits (862) a request for the travel data 36 along with the session ID to the server. On receiving (864) the data associated with the session ID from the server, client executable 32 decodes (866) the data and graphically displays (868) the decoded data in panel 74 (FIG. 5).

25 A number of embodiments of the invention have been described. Nevertheless, it will be understood that various modifications may be made without departing from the spirit and scope of the invention. For example, the client executable may be stored on a different server from travel module 34 and the web page 30c configured so that it causes the web browser to download the client executable from the different server. The server 10 may include multiple
30 processors to allow it to more effectively perform multiple tasks at the same time.

The invention may also be applied to other kinds of servers to perform functions that are not related to travel. For example, the invention may be applied to a server that distributes music data along with an executable client for playing the music data.

Accordingly, other embodiments are within the scope of the following claims.

WHAT IS CLAIMED IS:

- 1 1. A method comprising:
2 transmitting information by:
3 transmitting an executable to a remote location over a network;
4 generating data, wherein at least part of the data is generated while the
5 executable is being transmitted; and
6 transmitting the generated data to the remote location over the network.
- 1 2. The method of claim 1 wherein a client at the remote location receives the transmitted
2 executable and the generated data and the client includes a client processor that executes
3 the executable to process the data.
- 1 3. The method of claim 2 wherein the transmitted executable is received by the client before
2 the generated data is received and the executable causes the client computer to indicate
3 that client computer is waiting for the data.
- 1 4. The method of claim 2 wherein a first portion of the generated data is received by the
2 client before a second portion of the generated data and the executable causes the client
3 to indicate that the first portion has been received before the second portion is received.
- 1 5. The method of claim 2 wherein a first portion of the generated data is received by the
2 client before a second portion of the generated data and the executable causes the client
3 processor to process the first portion before the second portion is received.
- 1 6. The method of claim 1 wherein a first portion of the data is generated before a second
2 portion of the data and at least part of the first portion of the data is transmitted while the
3 second portion is being generated.
- 1 7. The method of claim 1 wherein the information is transmitted in response to a request for
2 information.
- 1 8. The method of claim 7 wherein the generated data is transmitted in response to a request
2 for data separate from the request for information.

- 1 9. The method of claim 1 wherein transmitting the information further comprises:
2 transmitting to the remote location a web page containing a directive to a web
3 browser to request the executable; and
4 receiving a request for the executable from the web browser, wherein the executable
5 is transmitted to the remote location in response to the request for the executable.
- 1 10. An article comprising a machine-readable medium which stores machine-executable
2 instructions operable to cause a machine to:
3 transmit information by:
4 transmitting an executable to a remote location over a network;
5 generating data, wherein at least part of the data is generated while the
6 executable is being transmitted; and
7 transmitting the generated data to the remote location over the network.
- 1 11. The article of claim 10 wherein the instructions cause the machine to generate a first
2 portion of the data before generating a second portion of the data and transmit at least
3 part of the first portion of the data while generating the second portion of the data.
- 1 12. The article of claim 10 wherein the instructions cause the machine to transmit the
2 information in response to a request for information.
- 1 13. The article of claim 10 wherein the instructions cause the machine to transmit the data in
2 response to a request for the data separate from the request for information.
- 1 14. An apparatus comprising:
2 a storage system which stores:
3 machine readable instructions, and
4 an executable; and
5 a server processor which executes the computer readable instructions to respond to a
6 request for information by:
7 transmitting the executable to a remote location over a network;
8 generating data, wherein at least part of the data is generated while the
9 executable is being transmitted; and

transmitting the generated data to the remote location over the
network.

15. The apparatus of claim 14 wherein the executable comprises instructions operable to
cause a client processor at the remote location to process the data.

16. The apparatus of claim 15 wherein the executable is received at the remote location
before the generated data is received and the executable causes the client processor to
indicate that client processor is waiting for the data.

17. The apparatus of claim 15 wherein a first portion of the data is received at the remote
location before a second portion of the data and the executable causes the client processor
to indicate that the first portion of the data has been received before the second portion is
received.

18. The apparatus of claim 15 wherein a first portion of the generated data is received at the
remote location before a second portion of the generated data and the executable causes
the client processor to process the first portion of the generated data before the second
portion is received.

19. The apparatus of claim 14 wherein a first portion of the data is generated before a second
portion of the data and at least part of the first portion of the data is transmitted while the
second portion is being generated.

20. The apparatus of claim 14 wherein the information is transmitted in response to a request
for information.

21. The apparatus of claim 14 wherein the generated data is transmitted in response to a
request for the data separate from the request for information.

ABSTRACT

A method includes transmitting information by transmitting an executable to a remote location over a network, generating data that is associated with the executable, and transmitting the generated data to the remote location over the network. At least part of the data is generated while the executable is being transmitted over the network.

5

20138889.doc

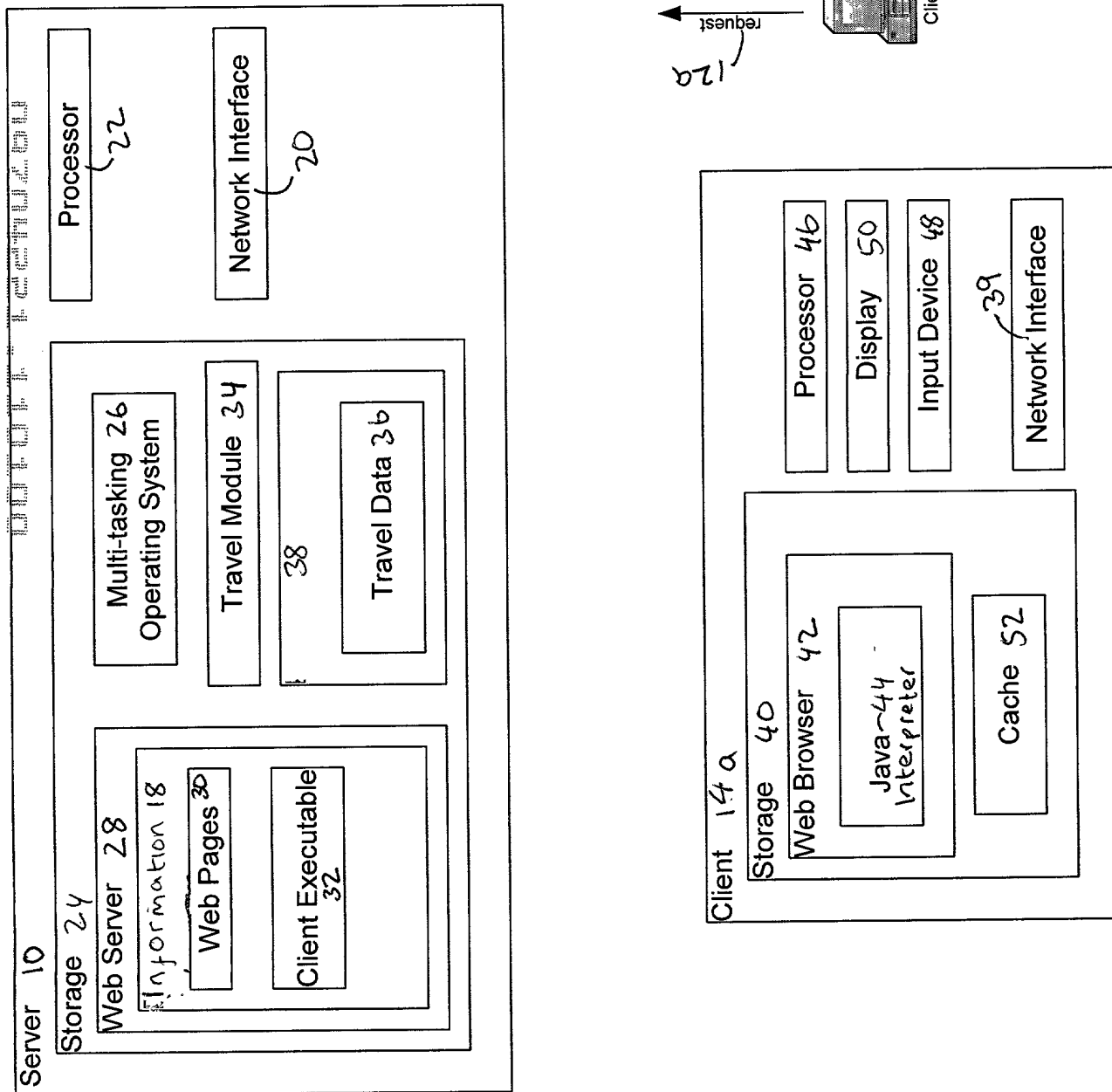


Fig. 1

ITA Software: Trip Planner - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Discuss Go Links

Address: http://beta.itsoftware.com/service/yodispach/ - 52

Flight Planning: 54a

From: bus Add airports within 50 miles

Depart: Nov 8 Leave Early Morning Alternate dates: none

To: sfg 5YC Add airports within 50 miles

Return: Nov 9 Leave Evening Alternate dates: none

Origins and destinations may be city or airport names or codes. Separate multiple entries with semicolons.

Passengers:

Adults: 12 to 61 (over 82) (2 to 11) (under 2) 1 0 0 0

Seniors: 0 0 0 0

Children: 0 0 0 0

Infants: 0 0 0 0

View results using (Help)

Trip Finder (try this first)

Flight Chooser (Java)

GO

54b

54c

54d

54e

54f

54g

54h

54i

54j

54k

54l

54m

54n

54o

54p

54q

54r

54s

54t

54u

54v

54w

54x

54y

54z

54aa

54ab

54ac

54ad

54ae

54af

54ag

54ah

54ai

54aj

54ak

54al

54am

54an

54ao

54ap

54aq

54ar

54as

54at

54au

54av

54aw

54ax

54ay

54az

54ba

54bb

54bc

54bd

54be

54bf

54bg

54bh

54bi

54bj

54bk

54bl

54bm

54bn

54bo

54bp

54bq

54br

54bs

54bt

54bu

54bv

54bw

54bx

54by

54bz

54ca

54cb

54cc

54cd

54ce

54cf

54cg

54ch

54ci

54cj

54ck

54cl

54cm

54cn

54co

54cp

54cq

54cr

54cs

54ct

54cu

54cv

54cw

54cx

54cy

54cz

54da

54db

54dc

54dd

54de

54df

54dg

54dh

54di

54dj

54dk

54dl

54dm

54dn

54do

54dp

54dq

54dr

54ds

54dt

54du

54dv

54dw

54dx

54dy

54dz

54ea

54eb

54ec

54ed

54ee

54ef

54eg

54eh

54ei

54ej

54ek

54el

54em

54en

54eo

54ep

54eq

54er

54es

54et

54eu

54ev

54ew

54ex

54ey

54ez

54fa

54fb

54fc

54fd

54fe

54ff

54fg

54fh

54fi

54fj

54fk

54fl

54fm

54fn

54fo

54fp

54fq

54fr

54fs

54ft

54fu

54fv

54fw

54fx

54fy

54fz

54ga

54gb

54gc

54gd

54ge

54gf

54gg

54gh

54gi

54gj

54gk

54gl

54gm

54gn

54go

54gp

54gq

54gr

54gs

54gt

54gu

54gv

54gw

54gx

54gy

54gz

54ha

54hb

54hc

54hd

54he

54hf

54hg

54hh

54hi

54hj

54hk

54hl

54hm

54hn

54ho

54hp

54hq

54hr

54hs

54ht

54hu

54hv

54hw

54hx

54hy

54hz

54ia

54ib

54ic

54id

54ie

54if

54ig

54ih

54ii

54ij

54ik

54il

54im

54in

54io

54ip

54iq

54ir

54is

54it

54iu

54iv

54iw

54ix

54iy

54iz

54ja

54jb

54jc

54jd

54je

54jf

54jg

54jh

54ji

54jj

54jk

54jl

54jm

54jn

54jo

54jp

54jq

54jr

54js

54jt

54ju

54jv

54jw

54jx

54jy

54jz

54ka

54kb

54kc

54kd

54ke

54kf

54kg

54kh

54ki

54kj

54kk

54kl

54km

54kn

54ko

54kp

54kq

54kr

54ks

54kt

54ku

54kv

54kw

54kx

54ky

54kz

54la

54lb

54lc

54ld

54le

54lf

54lg

54lh

54li

54lj

54lk

54ll

54lm

54ln

54lo

54lp

54lq

54lr

54ls

54lt

54lu

54lv

54lw

54lx

54ly

54lz

54ma

54mb

54mc

54md

54me

54mf

54mg

54mh

54mi

54mj

54mk

54ml

54mm

54mn

54mo

54mp

54mq

54mr

54ms

54mt

54mu

54mv

54mw

54mx

54my

54mz

54na

54nb

54nc

54nd

54ne

54nf

54ng

54nh

54ni

54nj

54nk

54nl

54nm

54nn

54no

54np

54nq

54nr

54ns

54nt

54nu

54nv

54nw

54nx

54ny

54nz

54oa

54ob

54oc

54od

54oe

54of

54og

54oh

54oi

54oj

54ok

54ol

54om

54on

54oo

54op

54oq

54or

54os

54ot

54ou

54ov

54ow

54ox

54oy

54oz

54pa

54pb

54pc

54pd

54pe

54pf

54pg

54ph

54pi

54pj

54pk

54pl

54pm

54pn

54po

54pp

54pq

54pr

54ps

54pt

54pu

54pv

54pw

54px

54py

54pz

54qa

54qb

54qc

54qd

54qe

54qf

54qg

54qh

54qi

54qj

54qk

54ql

54qm

54qn

54qo

54qp

54qq

54qr

54qs

54qt

54qu

54qv

54qw

54qx

54qy

54qz

54ra

54rb

54rc

54rd

54re

54rf

54rg

54rh

54ri

54rj

54rk

54rl

54rm

54rn

54ro

54rp

54rq

54rr

54rs

54rt

54ru

54rv

54rw

54rx

54ry

54rz

54sa

54sb

54sc

54sd

54se

54sf

54sg

54sh

54si

54sj

54sk

54sl

54sm

54sn

54so

54sp

54sq

54sr

54ss

54st

54su

54sv

54sw

54sx

54sy

54sz

54ta

54tb

54tc

54td

54te

54tf

54tg

54th

54ti

54tj

54tk

54tl

54tm

54tn

54to

54tp

54tq

54tr

54ts

54tt

54tu

54tv

54tw

54tx

54ty

54tz

54ua

54ub

54uc

54ud

54ue

54uf

54ug

54uh

54ui

54uj

54uk

54ul

54um

54un

54uo

54up

54uq

54ur

54us

54ut

54uu

54uv

54uw

54ux

54uy

54uz

54va

54vb

54vc

54vd

54ve

54vf

54vg

54vh

54vi

54vj

54vk

54vl

54vm

54vn

54vo

54vp

54vq

54vr

54vs

54vt

54vu

54vv

54vw

54vx

54vy

54vz

54wa

54wb

54wc

54wd

54we

54wf

54wg

54wh

54wi

54wj

54wk

54wl

54wm

54wn

54wo

54wp

54wq

54wr

54ws

54wt

54wu

54wv

54ww

54wx

54wy

54wz

54xa

54xb

54xc

54xd

54xe

54xf

54xg

54xh

54xi

54xj

54xk

54xl

54xm

54xn

54xo

54xp

54xq

54xr

54xs

54xt

54xu

54xv

54xw

54xx

54xy

54xz

54ya

54yb

54yc

54yd

54ye

54yf

54yg

54yh

54yi

54yj

54yk

54yl

54ym

54yn

54yo

54yp

54yq

54yr

54ys

54yt

54yu

54yv

54yw

54yx

54yy

54yz

54za

54zb

54zc

54zd

54ze

54zf

54zg

54zh

54zi

54zj

54zk

54zl

54zm

54zn

54zo

54zp

54zq

54zr

54zs

54zt

54zu

54zv

54zw

54zx

54zy

54zz

October 2000

Su Mo Tu We Th Fr Sa

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

November 2000

Su Mo Tu We Th Fr Sa

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

December 2000

Su Mo Tu We Th Fr Sa

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

January 2001

Su Mo Tu We Th Fr Sa

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

Legend: = holiday

Fig. 2

30a

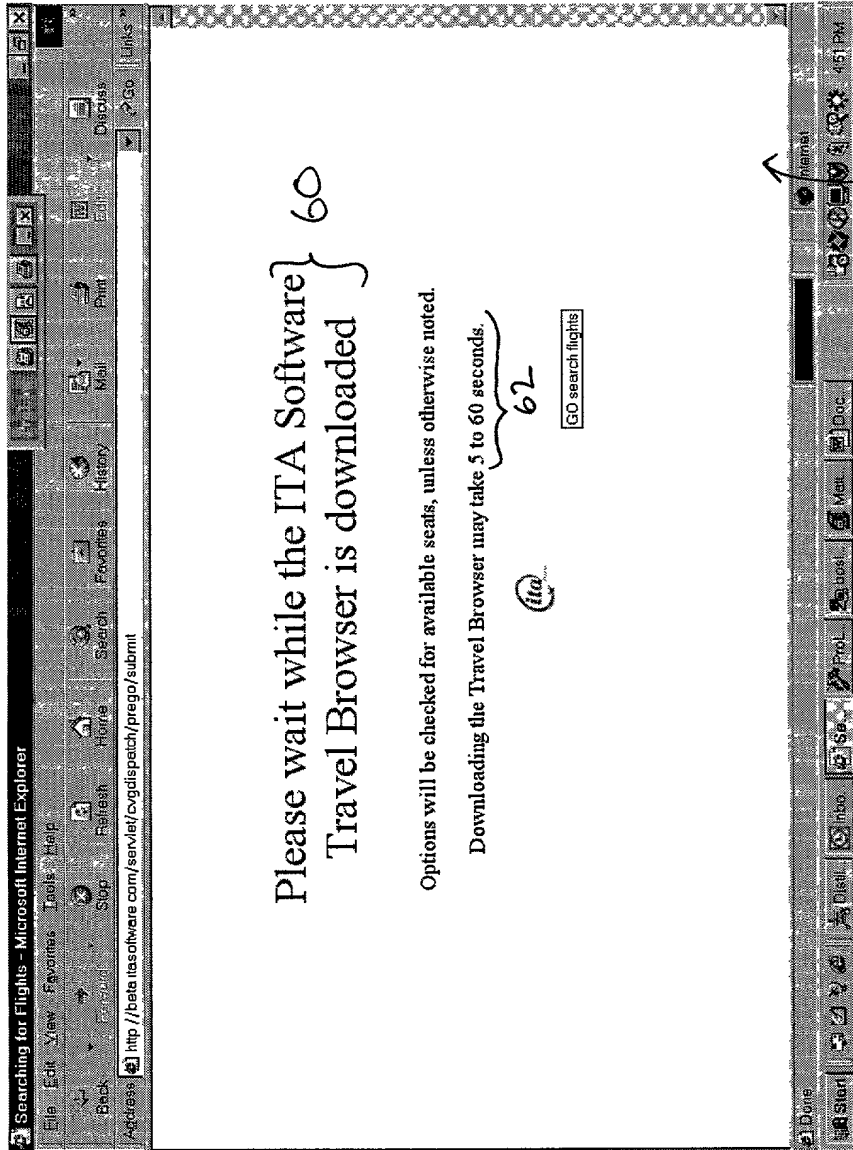


Fig. 3

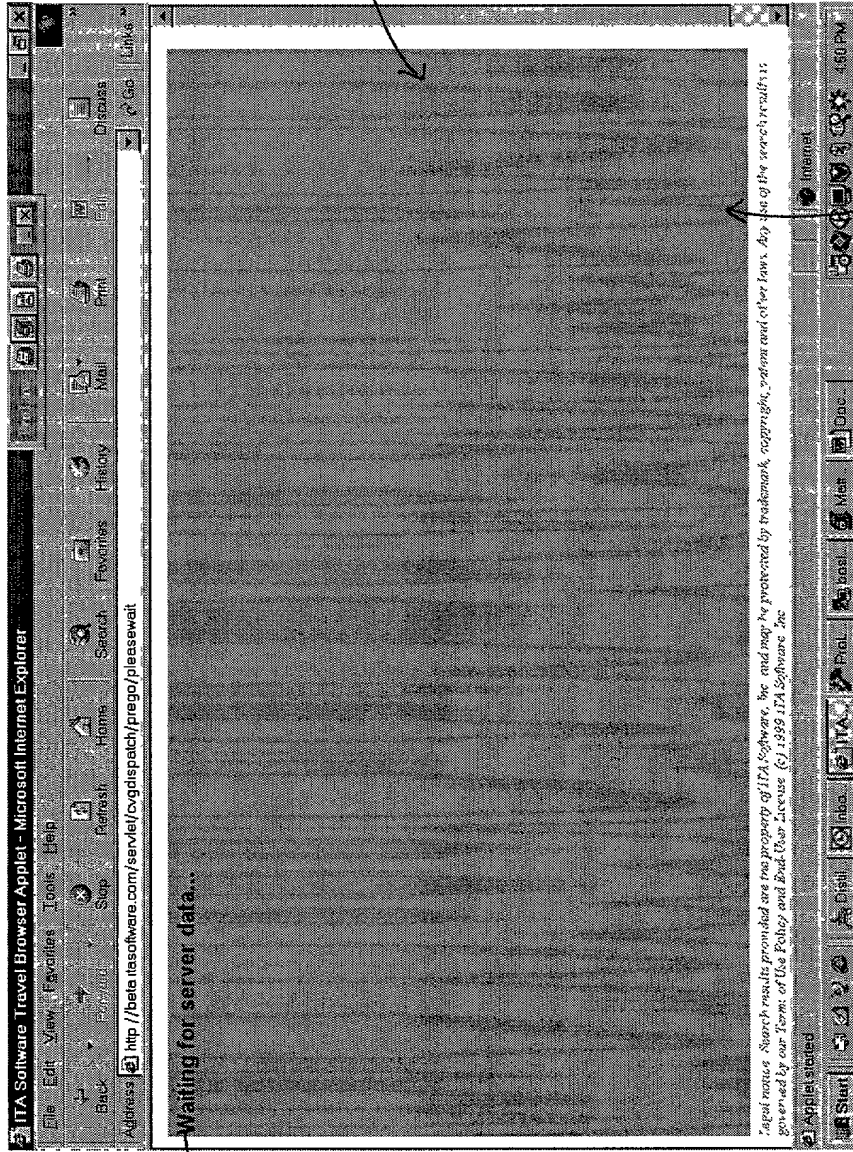
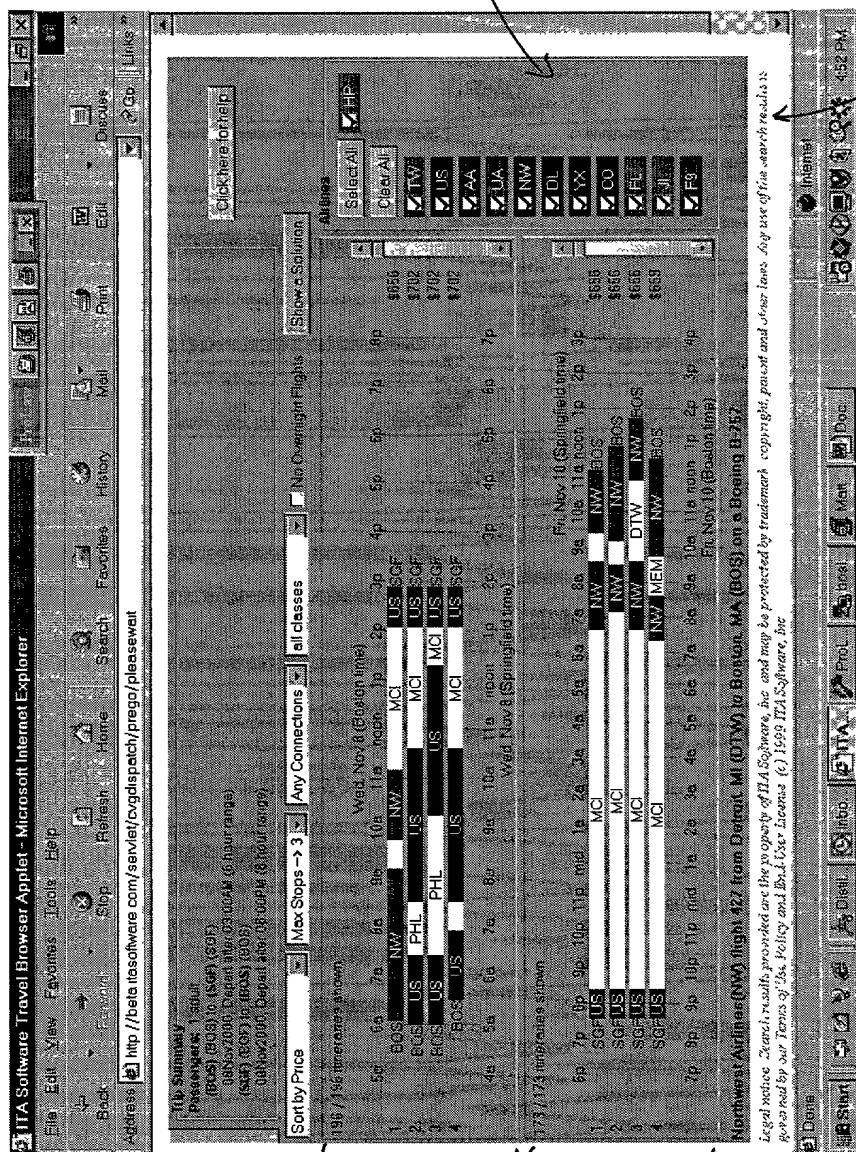
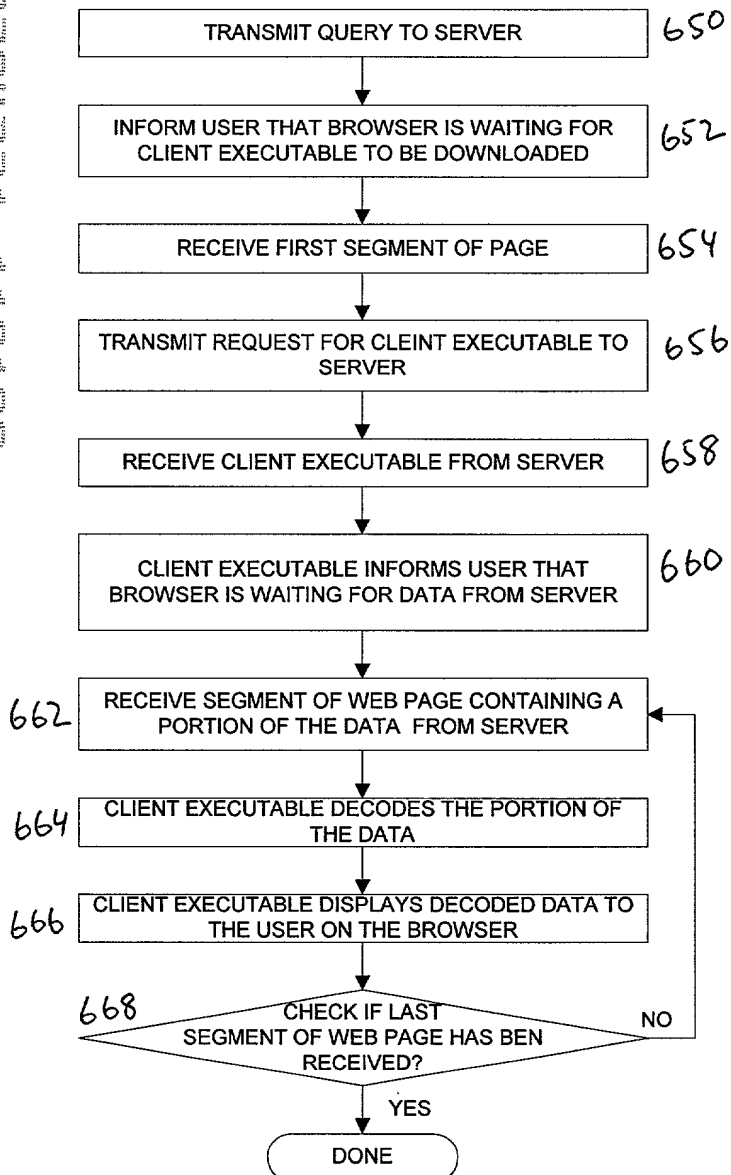
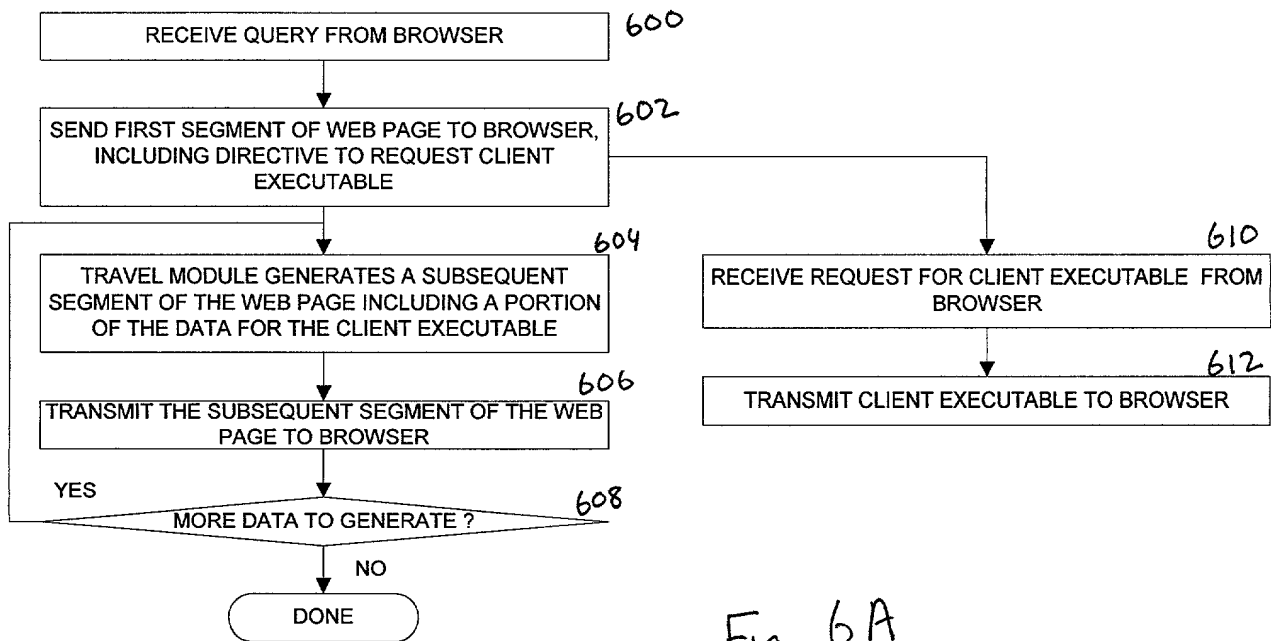


Fig. 4



5



```

<html>
<head><title> ITA Software Travel Browser Applet </title> </head>
<body>
<applet
  archive = /sproto/classes/Pogo85Applet.zip
    code = ITA.pogo85.Pogo85Applet
    width = 99% } 90
    height = 96% } 92
    hspace = 0 }
    vspace = 0 }
    name = qpmain>

<param name=pogocolor      value=#85ADAD>
<param name=logodir        value=/sproto/logos>
<param name=audiodir       value=/sproto/sounds>
<param name=helpurl        value=/servlet/cvgdispatch/help/hfd-intro>
<param name=atauid         value=971902950867>

</applet>

```

Handwritten annotations and groupings:

- A large bracket on the left groups the entire code block, labeled **80**.
- A bracket groups the `width = 99%` and `height = 96%` lines, labeled **90**.
- A bracket groups the `hspace = 0` and `vspace = 0` lines, labeled **92**.
- A bracket groups the `code = ITA.pogo85.Pogo85Applet` line and the `width`/`height` lines, labeled **88**.
- A bracket groups the `archive = /sproto/classes/Pogo85Applet.zip` line and the `code` line, labeled **84**.
- A bracket groups the `width`, `height`, `hspace`, and `vspace` lines, labeled **86**.
- A bracket groups the `archive` line, the `code` line, and the `width`/`height`/`hspace`/`vspace` lines, labeled **82**.
- A bracket groups the four `<param>` lines, labeled **94**.

Fig. 7 A

100 {

```

    <applet
      archive = "/sproto/classes/Pogo85Applet.zip"
      code = "ITA.pogo85.EmptyApplet"
      width = 0 } 104
      height = 0 }
      hspace = 0 } 106
      vspace = 0 }
      name = qpxdata0
      align = middle
    >

    <param name="qpxdata0"
      value="/Qt'FLX[99fhwcq?O,gUT?mw(1,]**zQ'ThQu@10Dc:>,f7>N6pRZQ3H_j
      7RuM2)vDFP`itTA1,M]ODm)PwU`Wssv7[:CG9f'm2nH9HGUYifhx=<+3aMuE>oz0<T
      evP7pkFrY`+<LPsl8N`r9PP0(t.,8<9C7_nc.NVYA6\Kk]drr7Iay<voPs'mpj)cDB
      Hum_{*RbMDz<IUdKRoH/?OJFL)zNa;qPCV?>g:D8hhxO>{zUHPqY9JtueJq)p[xAmg
      5F;JR)Rd+j)Z>:ZON8TY8`L^g)ACS*E\1*0z/<{zb-y;iO4TQpvNX81">
    </applet>
  
```

102 {

84

110 { 108

Fig. 7 B

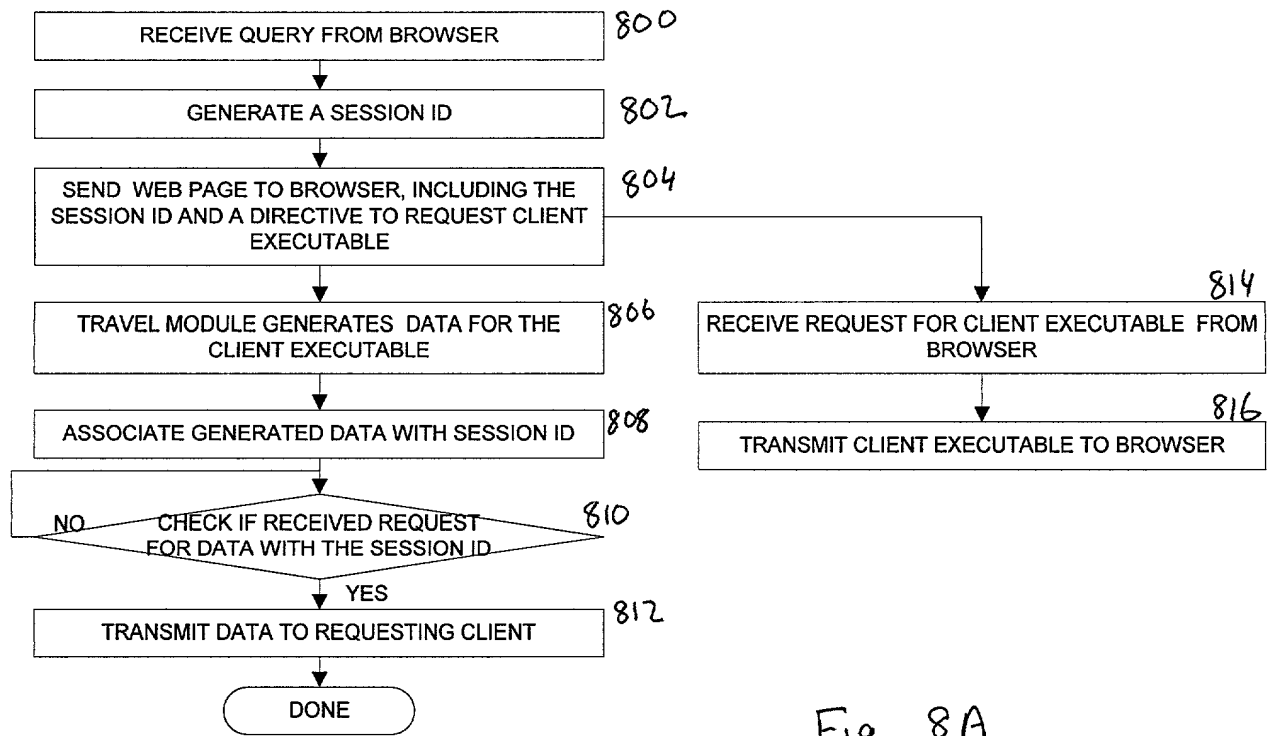


Fig. 8A

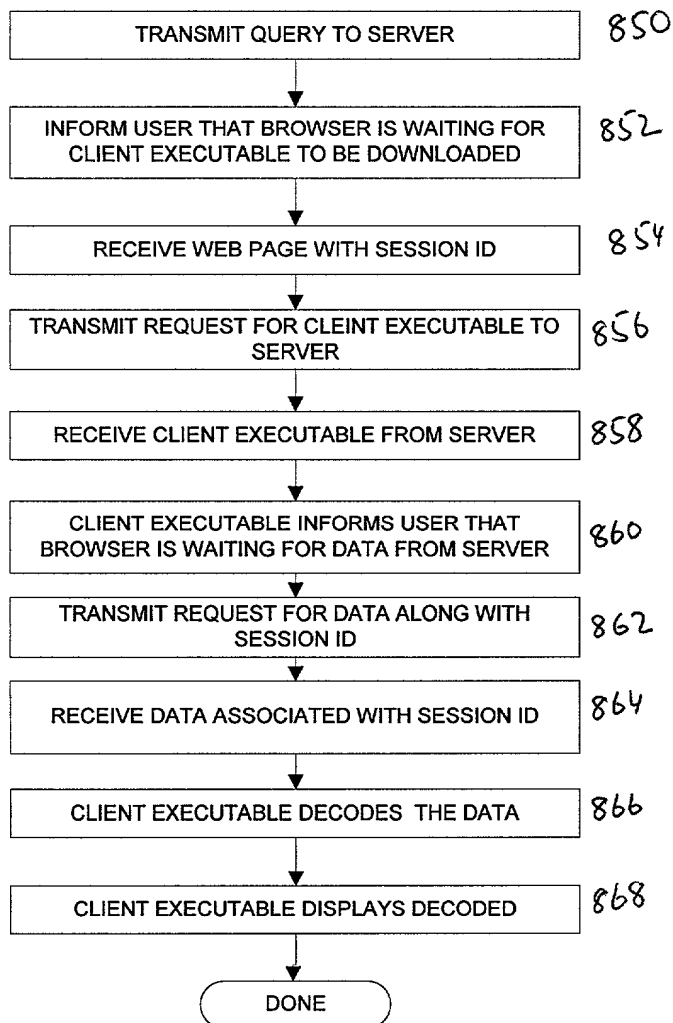


Fig. 8B